A Q-Learning approach with collective contention estimation for bandwidth-efficient and fair access control in IEEE 802.11p vehicular networks

Article (Accepted Version)

A Q-Learning Approach with Collective Contention Estimation for Bandwidth-efficient and Fair Access Control in IEEE 802.11p Vehicular Networks

Andreas Pressas, *Student Member, IEEE,* Zhengguo Sheng, *Senior Member, IEEE,* Falah Ali, *Senior Member, IEEE,* Daxin Tian, *Senior Member, IEEE,*

***Abstract*—Vehicular Ad hoc Networks (VANETs) are wireless networks formed of moving vehicle-stations, that enable safety-related packet exchanges among them. Their infrastructure-less, unbounded nature allows the formation of dense networks that present a channel sharing issue, which is harder to tackle than in conventional WLANs, due to fundamental differences of the protocol stack. Optimizing channel access strategies is important for the efficient usage of the available wireless bandwidth and the successful deployment of VANETs. We present a Q-Learning-based approach to wirelessly network a big number of vehicles and enable the efficient exchange of data packets among them. More specifically, this work focuses on a IEEE 802.11p-compatible contention-based Medium Access Control (MAC) protocol for efficiently sharing the wireless channel among multiple vehicular stations. The stations feature algorithms that "learn" how to act optimally in a network in order to maximise their achieved packet delivery and minimise bandwidth wastage. Additionally, via a Collective Contention Estimation (CCE) mechanism which we embed on the Q-Learning agent, faster convergence, higher throughput and short-term fairness are achieved.**

***Index Terms*—Vehicular Ad Hoc Networks, Machine Learning, Access Control, Fairness, IEEE 802.11p, Link Layer, CSMA.**

I. Introduction

Vehicle-to-Vehicle (V2V) communications enable the wireless ad hoc networking of moving vehicles within a Region of Interest (RoI), for safety message exchanges and other purposes. The key enabling technology, specifying the physical (PHY) and medium access control (MAC) layers of the V2V stack is IEEE 802.11p, which enables communications Outside the Context of a Basic service set (OCB) via the Dedicated Short Range Communication (DSRC) frequencies at 5.9 GHz. With DSRC specifying a 1-hop range of up to 1 km Line-of-Sight (LoS), wireless vehicular networks will have to accommodate many transmitting vehicle-stations within the range of each other. Additionally, with the Internet of Vehicles proposing an ever increasing amount of promising applications, novel protocols are needed to meet challenging demands not addressed by the conventional standard, since IEEE 802.11p belongs in the IEEE 802.11 family of protocols originally designed to be used in WLANs. The DSRC PHY and MAC must be scalable and it is expected that the stack often will have to manage 50-100 interconnected stations. The primarily one-to-many (broadcast) nature of transmissions for VANETs also presents some problems for the IEEE 802.11-inherited MAC layer which is not designed to accommodate broadcast traffic accordingly.

A MAC protocol defines the rules of how the various network stations access the shared channel to avoid packet collisions. The de-facto MAC layer used in IEEE 802.11p-based networks is implemented as a Carrier Sense Multiple Access (CSMA) algorithm, which is a distributed, contention-based MAC. It is a better idea from centralised solutions such as TDMA or FDMA [1], since these would require synchronisation among stations which is difficult to achieve in such mobile, infrastructure-less networks. But there is still space for improvement, especially when it comes to wireless vehicular networks which are unbounded, ad hoc networks with long one-hop transmission range, that allows them to become quite dense and congested in urban environments, leading to packet collisions. Every vehicle must maintain a relative standard of transmission frequency or else the rest of the vehicles in near proximity would not be aware of its existence. A vehicle-station's packets colliding and being dropped effectively mean that it is disconnected from the wireless vehicular network for the period of time that these packets are dropped, which may pose safety concerns.

A. Challenges and Objectives

Due to the safety nature of the packets exchanged via DSRC and their short temporal validity, the contention window (CW), defined by CSMA for the purpose of randomising the time of access to the channel among the various stations to avoid collisions, is kept small according to the IEEE 802.11p specification. Studies [2] [3] have shown that the small CW size is a main cause of packet collisions in DSRC-based networks, which cannot be eliminated by the IEEE 802.11p MAC as it is. Additionally, the IEEE 802.11 MAC has an intrinsic (short-term) fairness problem whereby stations cannot gain access to the wireless medium with equal probability under heavy traffic conditions [4], which can often be the case

This research was sponsored by The Engineering and Physical Sciences Research Council (EPSRC) (EP/P025862/1), Royal Society-Newton Mobility Grant (IE160920), the National Natural Science Foundation of China under Grant No. 61672082 and No. 61822101 and Beijing Municipal Natural Science Foundation No. 4181002.

A. Pressas, Z. Sheng and F. Ali are with the Department of Engineering and Design, University of Sussex, UK. E-mail: a.pressas, z.sheng, f.h.ali@sussex.ac.uk.

D. Tian is with School of Transportation Science and Engineering, Beihang University, Beijing 100191, China. E-mail: dtian@buaa.edu.cn. (Corresponding Author)

in wireless vehicular networks. Unfair access opportunities could impair the reliability of critical applications as well as affect the quality of service (QoS) support for DSRC-based networks.

Applications for VANETs vary a lot, as do their communication requirements. Pre-crash sensing or (semi) autonomous applications such as Cooperative Adaptive Cruise Control (CACC) [5] rely on ultra-low latency exchanges ($\leq$ 20 ms) for warnings or directly driving vehicle control systems. Others are oriented towards more assistive, road safety and traffic efficiency uses such as lane-changing and emergency braking, with strict but more easily met latency requirements ($\leq$ 100 ms) [6]. Finally there are also convenience and infotainment uses where delay is not as critical in comparison but the transferred data volume can be much larger.

The objective of this paper is to develop a DSRC-compatible MAC layer capable of self-improving over time, that can meet key requirements for various VANET applications, such as reliability and bandwidth efficiency and low latency as well as enhancing short-term fairness and handling of service separation.

B. Contributions

We propose a self-learning channel sharing control method that can be biased towards satisfying various V2V applications, for both unicast and broadcast V2V exchanges via DSRC links. It allows to directly interconnect a big number of vehicles and stationary units via IEEE 802.11p wireless interfaces, by employing a Reinforcement Learning (RL) algorithm to perform CW adaptation. This technique allows the designers to improve networking performance via self-learning channel access controllers, without having to make major modifications to existing hardware. Moreover, the real-time learning and control requirements of the algorithm are considered. We suggest a strategy to handle the exploration-exploitation trade-off in a way that accelerates convergence and yields performance benefits within short time.

Furthermore, we design a novel Q-Learning reward mechanism with the ability to collectively estimate the a near-optimum system-wide CW, aiming to enhance bandwidth efficiency and mitigate the fairness problem. It is based on the fact that the CW size represents the contending priority and that the fairness issue can be tackled by adjusting the stations' CW size. Thus we propose a Collective Contention Estimation (CCE) algorithm inspired by [4] [7], that takes advantage of overheard transmissions made by contending nodes and biases the stations in the network towards contenting fairly by using similar CW values. This novel CCE technique for Q-Learning agents enables the design of a MAC protocol that progressively learns how to achieve fairness among all contenting links, and allows IEEE 802.11p to extend beyond its traditional basic safety message exchange capabilities by offering even higher throughput, enhanced fairness and better handling of simultaneous applications. Additionally, this reward mechanism assists the convergence of the proposed learning MAC algorithm towards a near-optimal system-wide CW value.

Finally we suggest a way of combining multiple sub-goals that should be met by the Q-Learning agents. This way the proposed Q-Learning based MAC protocol offers considerable advantages for deployment in IEEE 802.11p-based vehicular networks, regarding both throughput and fairness while also being able to satisfy low-latency requirements.

The remainder of the paper is organised as follows: Section II is a summary of the technologies that are studied and improved upon in this work. Section III is focused on protocol design intrinsics for improving V2V performance. Section IV is the performance evaluation of the proposed solutions, regarding throughput, fairness and delay. Finally Section V concludes our findings and suggests improvements for future work.

II. Literature Review

The IEEE 802.11 family of standards defines the MAC and physical layer protocols for implementing wireless local area networks (WLANs). The standards feature a Distributed Coordination Function (DCF) for sharing access to the common medium among multiple peers in a distributed manner. Li, *et al*. [8] studies the sensitivity of throughput, latency and fairness to changes of the CW_{min}, CW_{max} parameters of the DCF in IEEE 802.11-based networks with many contending stations. Modifications to the IEEE 802.11 DCF have been proposed regarding mitigating the inherent fairness problem of the DCF, such as the solutions presented in [4] and [7] which both use a $backoff$ copying scheme to achieve fairer bandwidth allocation among stations. However, traditional IEEE 802.11-based networks require that stations are interconnected via an Access Point, and are designed for unicast exchanges. Consequently the protocol cannot be used as-is for V2V communications, which has to be infrastructure-less and accommodate geo-significant transmissions to be received by all peers within a RoI.

The IEEE 802.11p (DSRC) amendment is proposed to tackle peer-to-peer (ad hoc) networking for vehicles. The MAC layer of the protocol adopts the DCF and includes the new OCB mode of operation which allows vehicles to form ad hoc networks among them and enable broadcast transmissions as the primary form of communication. Lu, *et al.* [9] identify the poor performance of the DSRC MAC in supporting safety applications mainly due to the high collision probability of the broadcasted packets as a key issue in the MAC layer of vehicular networks, while Xu, *et al*. [1] find TDMA is not appropriate to resolve the MAC issues presented. Campolo, *et al*. in [10] show that packet delivery probability, modelled as a function of CW and the number of vehicles, is negatively affected as the nodes increase. Then in [11] they suggest that increasing the CW size reduces the frame loss probability in a similar IEEE 802.11p broadcasting scenario. Kloiber, *et al*. [12] suggest that a bigger CW favours packet delivery for status-message broadcasting which is more delay-tolerant. Hassan, *et al*. in [13] also show the impact that vehicular density and increased traffic have on transmission reliability, in terms of packet delivery rates. Additionally, it proposes a new MAC protocol that trades increased packet delay, which still remains below the required threshold for most safety applications, for decreased packet loss by introducing retransmissions.

These findings contradict the analysis presented in [14] which suggests that big CW values will increase delay to the point that they can harm some V2V applications. Additionally, [6] shows that some proposed safety applications such as Pre-Crash Sensing / Cooperative Collision Mitigation cannot tolerate more than 20 ms of packet delivery latency. Wu, *et al.* in [15] employ a swarming approach for CW adaptation, towards optimising the one-hop delay in inter-platoon V2V communications. We conclude that there cannot be a value of CW that is suitable for all circumstances, and that can be a problem in broadcast IEEE 802.11p where by default the size of the parameter is not adapted to network traffic.

There has been emerging work on employing Reinforcement Learning towards handling the channel access control problem in wireless networks. Amuru, *et al* in [16] formulate the problem of optimizing the IEEE 802.11 $backoff$ mechanism as an MDP, and propose Reinforcement Learning algorithms as a solution. Liu, *et al*. in [17] adopt Reinforcement Learning as an energy-efficient channel sharing technique for wireless sensor networks. Wu, *et al*. in [18] propose a Q-Learning based MAC protocol for unicast, delay-sensitive VANET exchanges. We found that this work does not consider the broadcast nature of VANETs, or the learning algorithm convergence and real-time requirements set by such vehicular use-cases. Additionally there is a potential to further improve the performance regarding packet delivery for various latency requirements and fairness.

As a solution we design and present an IEEE 802.11p-compliant MAC algorithm based on Q-Learning. It simultaneously targets reliable packet delivery and throughput-fairness, while being latency-aware. It fully supports and enhances classic broadcast V2V systems by using implicit ACKs piggybacked in broadcast packets for added reliability and feedback for Q-Learning. It features the proposed CCE reward method for Q-Learning, designed to tackle the inherent fairness problem appearing in CSMA-based IEEE 802.11p networks, to achieve more efficient channel sharing in terms of providing (near) equal transmission opportunities and improved transmission reliability for all stations.

III. Preliminaries

A. IEEE 802.11p and CSMA

The work in this paper focuses on studying and improving the DCF, which is the contention-based protocol used for channel sharing in IEEE 802.11-based wireless networks. It employs the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm to manage access to the medium among stations in a distributed way. The protocol also features the Binary Exponential Backoff (BEB) algorithm, which has the ability to reduce collisions by reacting to increases in network traffic.

According to CSMA/CA, when there is a packet ready to transmit and the wireless medium is continuously found idle for a DCF Interframe Space (DIFS), the sending station randomly draws an integer $backoff$ from the uniform distribution over the interval $[0, CW]$ and counts down after every time slot while medium is still found idle. If the medium becomes busy, the station has to wait again for an Arbitration Inter-Frame Spacing (AIFS) before being able to continue decrementing the $backoff$ counter. When the $backoff$ reaches the value of 0, the packet is transmitted. This way, the IEEE 802.11 DCF mechanism at the MAC layer randomizes the time interval between two consequent transmissions on a specific channel. This reduces the probability of two stations transmitting simultaneously, which will lead to a collision and both packets being corrupted or completely dropped.

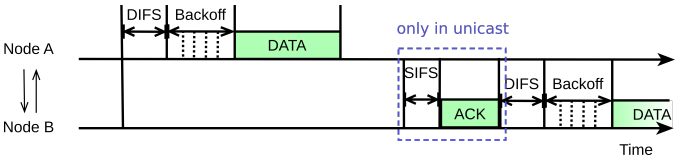

Fig. 1: A CSMA/CA cycle of operation, managing channel access among transmitting nodes A and B

The BEB algorithm, if enabled, adjusts the CW parameter based on the number of consecutive collisions detected by lack of incoming ACK packets. According to the algorithm, the CW value is doubled every time a packet collides, up to a CW_{max} quantity. When a transmission succeeds CW is reset to CW_{min}, from where it starts again for the next transmission.

When it comes to the IEEE 802.11p amendment for Vehicle-to-Vehicle communication, the BEB part of the DCF can be considered harmful since it relies on explicit ACK packets to adjust the $backoff$ parameter depending on whether a transmission was successful or not. This can cause increased delays and unreliability because the non-reception of ACK packets will block other urgent transmissions, as seen in [19]. Additionally, implementation of neither the BEB nor ACKs is done for broadcast (OCB) transmissions because they will cause the ACK implosion phenomenon [20] which can lead to service disruption, since there can be many recipients that will all return an ACK upon reception, causing more collisions and packet drops than actually help resolve network traffic congestion. This means that broadcast communication in DSRC has no acknowledgement feature and the choice of $backoff$ values is always limited within $[0, CW_{min}]$.

A small CW_{min} means that the stations will not have to wait for many time slots before they can transmit when the channel is sensed to be idle, which is good in sparse networks since it keeps the total transmission delay low and it helps not miss transmission opportunities because of waiting longer than needed. But in an urban environment where multiple vehicle-stations continuously transmit using a small CW_{min}, the probability of two or more stations drawing the same $backoff$ after both finding the channel idle and attempting to transmit simultaneously will unavoidably increase, which leads to packet collisions and bandwidth wastage.

Furthermore, the BEB mechanism presents an intrinsic fairness problem, because each station relies on its own limited experience to estimate congestion, which often leads to asymmetric views. Consequently, when the mechanism is utilised under high traffic loads, some nodes achieve significantly larger throughput than others, as shown in some studies in

literature [4] [21]. The problem occurs due to the fact that BEB resets the CW of a successful sender to CW_{min}, while other stations could continue to maintain larger CW sizes, thus reducing their chances of capturing the channel and resulting in continuous channel domination by the successful station. But even with the mechanism disabled, the big number of collisions in a congested wireless vehicular network can result in unfairness in the system. Consequently an efficient replacement that adapts the CW as needed to tackle the described packet drop and fairness problems could be of great use in such environments.

B. Q-Learning in Markovian Environments

There is a clear trade-off when selecting the CW size, since it should be large enough to be able to accommodate the network traffic as much as possible without collisions, but not unnecessarily large so that it increases packet transmission latency and stations miss opportunities to transmit because of waiting too long and the channel turning busy. For these reasons we employ Q-Learning to adapt the CW as needed. This algorithm requires insignificant computational capability from the MAC controller and has minimal networking overhead, apart from some form of reception acknowledgement that is typically standard in unicast wireless networks for reliability purposes and is utilised by most applied contention-based MAC protocols for the purpose of feedback.

The Markov Decision Process (MDP) formalism can be used to mathematically model Reinforcement Learning agents. An MDP is defined as a (S, A, P, R) tuple, where S stands for the set of possible states, A_s is the set of possible actions from state $s \in S$, $P_a(s, s')$ is the probability to transit from a state $s \in S$ to $s' \in S$ by performing an action $a \in A$. $R_a(s, s')$ is the reinforcement (or immediate reward), resulting from the transition from state s to state s' because of an action a. The decision policy π maps the state set to the action set, $\pi : S \rightarrow A$. Therefore, the MDP can be solved by discovering the optimal policy that decides the action $\pi(s) \in A$ that the agent will make when in state $s \in S$.

In practical scenarios such as the channel sharing problem studied here, though, the transition probability $P_{\pi(s)}(s, s')$ is rarely known, which makes it difficult to evaluate the policy π. Q-Learning [22] [23] is an effective model-free learning algorithm, used to find (near) optimum solutions π for MDPs from delayed reinforcement, without knowing $P_{\pi(s)}(s, s')$ a-priori. It essentially provides agents the ability to learn how to behave optimally in Markovian domains by experiencing the consequences of their actions, without requiring maps of these domains.

A Q-Learning agent maintains a table of $Q[S, A]$, where S is the set of states and A is the set of actions. At each discrete time step $t = 1, 2, \ldots, \infty$, the agent observes the current state $s_t \in S$ of the MDP, selects an action $a_t \in A$, receives the resulting reward r_t and then observes the next state $s_{t+1} \in S$ it transitions to because of that action a_t. This sequence of events is a learning experience (s_t, a_t, r_t, s_{t+1}) for the agent, which updates the Q-table at the observed state-action pair according to function (1). Essentially, the algorithm is based on value iteration update. It tries to correctly calculate the quality of a state-action (s, a) combination $Q(s_t, a_t)$ by assuming the current value and making a correction based on the newly acquired information, as seen in (1). The goal of the agent is to maximise its acquired cumulative reward over time.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times [r_t + \gamma \times \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \; - Q(s_t, a_t)] \tag{1}$$

The discount factor γ models the importance of future rewards. Setting $\gamma = 0$ will make the agent "myopic" or short-sighted by only considering current rewards, while setting it close to $\gamma = 1$ will make the agent strive for a high long-term reward. Usually this parameter is set between 0.6 to 0.99, and is considered to be part of the problem. The learning rate α quantifies to what extent the newly acquired information will override the old information. An agent with $\alpha = 0$ will not learn anything new, while with $\alpha = 1$ it considers only the most recent information. The $\max_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1})$ quantity is the maximum Q-value across all actions a which are possible at the next state s_{t+1}.

IV. Q-Learning MAC Protocol Design

The adaptive backoff problem fits into the MDP formulation. RL is used to design a MAC protocol that selects the appropriate CW value based on gained experience from its interactions with the environment within an immediate communication zone. The proposed MAC protocol, features a Q-Learning-based algorithm that adjusts the CW size based on feedback given from probabilistic rebroadcasts in order to avoid packet collisions. In the remaining of this section we present employing (1) as a learning, self-improving, control protocol for sharing the wireless medium among multiple IEEE 802.11p stations. The protocol basically works as follows; a station transmits a packet and then gets feedback r_t depending on the outcome of this transmission, determined by the reception or not of a packet containing an ACK within an acceptable Round-Trip Time (RTT). The Q-Learning agent then adapts the station's CW value accordingly before sending the next packet, and then the process is repeated. The baseline Q-Learning MAC protocol's operation is depicted in Fig. 2.

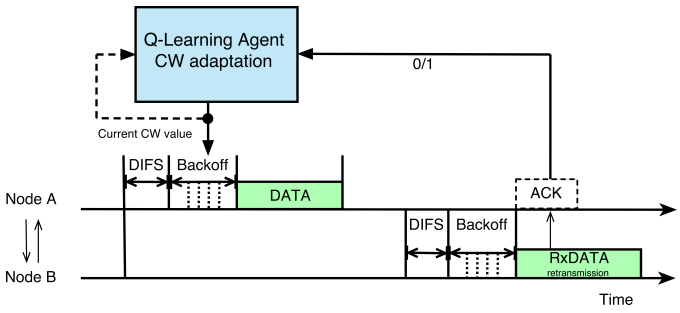

Fig. 2: Baseline Q-Learning based MAC protocol

By employing the logic behind $backoff$ copying [4] [7] combined with internal critics that provide goal-specific "advice" [24] in the form of state-dependent rewards, we can

enhance the Q-Learning MAC protocol to satisfy various kinds of V2V applications. This way different reward functions can be developed and utilised depending on whether we strive for high reliability (packet delivery) and fairness or low latency or even balancing both.

Algorithm 1 Q-Learning MAC

Initialize $Q_0[CW, A]$ at $t_0 = 0$
$CW_0 = CW_{min} = 3$ at $t_0 = 0$
if $N_{tx} < N_{train}$ **then**
    $\varepsilon, \alpha \leftarrow decay\,function$ ▷ according to rule (4)
else
    $\varepsilon, \alpha \leftarrow constant$
end if

procedure ACTION_SELECTION(CW_t) ▷ ε-greedy
    if $p_\varepsilon \leq \varepsilon$ **then**
        $a_{t+1} \leftarrow random(^{(CW_t\,-\,1)}/_2, CW_t, CW_t * 2 - 1_)$
    else if $p_\varepsilon \geq 1 - \varepsilon$ **then**
        $a_{t+1} \leftarrow a_\pi$
    end if
    $CW_{t+1} \leftarrow CW^{a_{t+1}}$
end procedure

procedure SEND(TxPacket, SeqID, CW_{t+1})
    TxPacket**.setOriginId**(SeqID)
    $RTT \leftarrow 0\,s$
    Content(CW_{t+1})
    $CW_{t-1} \leftarrow CW_t$
    $CW_t \leftarrow CW_{t+1}$
end procedure

procedure FEEDBACK(CW_t, CW_{t-1},RxPacket)
    if RxPacket**.GetOriginId=** TxPacket**.GetSeqId**
        && $RTT < 0.1$ s **then**
        if $a_t \neq (CW_t \longleftarrow CW_{t-1})$ **then**
            $r_t \leftarrow R_{func}(CW_t)$
        end if
    else if $RTT = 0.1$ s **then**
        $r_t \leftarrow -1$
    end if
    update $Q(CW_{t+1}, a_{t+1})$ ▷ according to rule (1)
    Action_selection(CW_t)
end procedure

A. On-line Decision Making Dilemma

The Q-Learning algorithm's primary purpose in this application is to converge to a (near) optimum output, in terms of packet delivery reliability. It achieves this by transitioning to different CW values (states S) by performing actions $a \in A$, transmitting packets and then getting experience from these transmissions using said CW values, via feedback in the form of overheard retransmissions. The operation of the proposed self-learning channel access control mechanism is summarised in Algorithm 1.

Watkins, *et al* [22] proved that Q-Learning converges to the optimum (s, a) pair/s with probability 1 as long as all actions are repeatedly sampled in all states s and the (s, a) pairs are represented discretely. To meet the second convergence criterion, the explored state space S contains 7 discrete IEEE 802.11p-compliant CW values ranging from 3 to 255. The CW is adapted according to (2), prior to every packet transmission attempt. The action space A contains the 3 following actions a, which are the same the BEB mechanism uses to adapt the CW upon transmission failure.

$$CW_{t+1} \xleftarrow{a \in \{^{CW_t\,-\,1}/_2, CW_t, CW_t * 2 - 1\}} CW_t \tag{2}$$

RL algorithms differ from supervised learning [25] ones in that correct input-output pairs are never presented, and sub-optimal actions are not explicitly corrected. In addition, there is a focus on on-line performance, which necessitates finding a balance between exploration of uncharted territory and exploitation of already acquired knowledge. This in practice translates as a trade-off in how the learning agent in this protocol selects its next action for every algorithm iteration. It can either explore by randomly picking an action from (2) so that the algorithm can transit to a different (s, a) pair and get experience (reward) from it, or follow a greedy strategy that exploits its so-far gained experience, and choose the action a which yields the highest Q-value for the state s it is currently in, given by

$$\pi(s) = \arg\max_a Q(s, a)\,. \tag{3}$$

The greedy strategy with respect to the Q-values tries to exploit continuously, however, since it does not necessarily explore all (s, a) pairs properly, it fails satisfying the first convergence criterion. On the other side, a fully random policy continuously explores all (s, a) pairs, but it will behave sub-optimally as a controller. An interesting compromise between the two extremes is the ε-greedy policy [26], which executes the greedy policy with probability $1 - \varepsilon$. This balancing between exploitation and exploration can guarantee convergence and yield good performance.

B. Decaying ε-greedy strategy

In practice the Q-Learning algorithm converges under different factors depending on the application and complexity. The proposed protocol uses the ε-greedy strategy to focus the algorithm's exploration on the most promising CW trajectories. This strategy can guarantee the first convergence criterion by forcing the agent to sample all (s, a) pairs over time with probability ε. Consequently, the proposed algorithmic implementation satisfies both convergence criteria, but further optimisation is needed regarding convergence speed and applicability of the system.

The $Q[CW, A]$ table with size $[7, 3]$ is initialized to zero, except from $Q[0, 0]$ and $Q[6, 2]$ which are set to extreme negative values (i.e., -100), since they should never be visited by the agent and practically bound the CW size. When deployed in a new environment (initialised $Q[S, A]$), the agent should mostly explore and value immediate rewards, and then progressively show its preference for the discovered (near) optimal actions $\pi(s)$ as it is becoming more sure of its Q estimates. This can be achieved via the function shown below,

$$\varepsilon = e^{-\lambda * \frac{N_{tx}}{N_{train}}} \quad \text{for} \quad 0 \leq N_{tx} \leq N_{train}, \tag{4}$$

where N_{tx} is the number of transmitted broadcast packets and N_{train} is a pre-set number of packets that sets the training (decay) period. This strategy is based on the ε-greedy strategy, however the ε value decays over time instead of being constant. The strategy starts with a high $\varepsilon = 1$, and thus only explores by only performing random actions trying to fill the Q-table for all (s, a) pairs. Over time ε becomes progressively smaller until it fades (or reaches a minimum value), as we trust that the algorithm has converged to the optimal π, so that this learnt π can be executed without performing more (possibly sub-optimal) exploratory actions. In our implementation the ε value decreases as a function of the number of transmitted packets since the agent's deployment, until it reaches a minimum value of $\varepsilon = 0.05$ which essentially makes the agent perform a random action just 5% of the time for the purpose of self-correction even when used as a controller.

Additionally, reducing the value of α over time via the same function (4), essentially forces the agent to progressively limit the rate of overriding the existing experience by newly acquired rewards. This way, the so-far found (near) optimal states-CW/s are revealed as the agent becomes more confident in its so-far gained experience as time progresses, and behaves better as a controller avoiding big oscillations around the CW value yielding the highest cumulative reward. We propose that both quantities undergo exponential decay, rather than linear decay since it forces the system to use gained experience and limits randomness much faster, which is especially useful for mobile environments such as vehicular networks. A larger decay constant λ will make ε and α vanish more rapidly, which may negatively affect learning.

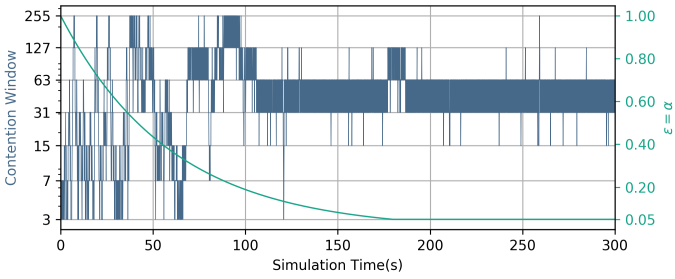

Fig. 3: CW adaptation by a single station utilising our Q-Learning based MAC protocol

C. Reward Function Formulation

An RL agent receives positive or negative reinforcement in the form of a scalar reward signal upon acting so that it can learn to behave correctly in its environment. Taking advantage of the full channel capacity and achieving maximum packet delivery (throughput) is of primary concern for this system, aiming to satisfy the reliability requirement of V2V traffic. This can be accomplished by employing a simplistic binary reward function according to which the agent is rewarded with 1 in case of successful transmission (ACK received) and -1 in the case of a failed transmission, first presented and evaluated in [27]. Fig. 3 shows the operation of this protocol regarding CW adaptation over time, alongside the decaying $\varepsilon - greedy$ function.

In this design the the agent has a harder problem to solve, compared to using a more detailed reward function, where there is a reward gradient over states. Specifying a more detailed reward function can help the algorithm converge faster, since more clues are provided. Evaluative feedback from internal critics associated with specific goals can be employed to make a function which returns a different reward depending on the CW that was used for every transmission, leading to faster convergence as well as better networking performance. Essentially we can bias the Q-Learning agent to prefer some CW values instead of others, depending not only on the success of transmission but also on (a set of) sub-goals which optimise some other performance-related objective(/s).

Based on this logic, we present a gradient-based reward function designed for the needs of urban vehicular networks where bandwidth efficiency and fairness regarding channel occupation among stations are of utmost importance. It is based on copying the CW sizes used by neighbouring transmitting stations and comparing them with the CW the on-board Q-Learning agent suggests. The reward is based both on the success of the packet and the result of that comparison. This addition can be utilised when having many vehicles with similar network presence (i.e. data rate, number of transmitting neighbouring stations) and helps to collectively find the optimum CW that accurately reflects the level of contention. We also validate the delay-sensitive scheme found in [18] and propose a function that combines both sub-goals together.

1) Collective Contention Estimation (CCE): Inspired by [4] [7], we adapt and introduce the $backoff$ copying idea to the Q-Learning agent, in which the receiving stations copy the CW size from overheard data frames coming from nearby stations that experience similar network conditions. This technique can be used as a way to bias the reward function so that agent-stations collectively estimate the network congestion level, as well as compete more fairly for the channel, since all of them content with fairly similar CW sizes.

Our mechanism starts with a piggybacking routine in which the employed CW value for each transmission is piggybacked onto the packet to be transmitted. Receiving stations invoke a CW copying routine, which adds the CW value to a $\Sigma CW[]$ vector. The size of the vector depends on the number of receipt transmissions and a set $PacketsWindow$ parameter. Once the vector fills up, for every new added CW value the last one is removed (FIFO). That way every agent utilising this algorithm considers only the latest receipt CW values, which helps estimate the network-wide congestion level for as long as the window dictates (1 second in this case to keep up with increased mobility and changing topology of vehicular networks).

We use the term "popular" for a CW size, by meaning that the receiving station notices that other transmitters often achieve successful transmissions when using it. A CW size is the most popular system-wise when used for the majority

Algorithm 2 Collective Contention Estimation Algorithm

$\Sigma CW = []$
$CW_{levels}[7] = [3, 7, 15, 31, 63, 127, 255]$
$Reward_{CW}[7] = [1/7, 2/7, 3/7, 4/7, 5/7, 1]$

procedure CW_COPY(RxPacket)
  if $RxPacket$.**GetAppType** $= Self$.**GetAppType**
    && $RxPacket$.**GetExplore** $= 0$ **then**
    $PacketsWindow++$ ▷ Resets to 0 every 1s
    if $length(\Sigma CW) > PacketsWindow$ **then**
      $\Sigma CW[]$.**remove**$(\Sigma CW[0])$
    end if
    $\Sigma CW[]$.**add**$(Packet.GetCW)$
  end if
end procedure

procedure R_{CCE}(RxPacket)
  for $i \leftarrow 0; i < length(CW_{levels[]}); i++$ **do**
    if $RxPacket$.**GetCW** $= CW_{levels}[i]$ **then**
      $index_{CW} \leftarrow i$ ▷ Find CW index
    end if
  end for
  $counter_{CW} \leftarrow 0$
  for $i \leftarrow 0, i < length(\Sigma CW[]), i++$ **do**
    if $\Sigma CW[i] = RxPacket$.**GetCW then**
      $counter_{CW}++$
    end if
  end for
  $Frequencies_{CW}[index_{CW}] \leftarrow \frac{counter_{CW}}{length(\Sigma CW[])}$
  $SortedFrequencies_{CW}[] \leftarrow Frequencies_{CW}[]$
  sort$(SortedFrequencies_{CW}[])$
  for $i = 0; i < length(Frequencies_{CW}[]); i++$ **do**
    if $Frequencies_{CW}[index_{CW}] =$
    $SortedFrequencies_{CW}[i]$ **then**
      $index_{reward} \leftarrow i$
    end if
  end for
  Return $Reward_{CW}[index_{reward}]$
end procedure

of (successful) overheard transmissions from stations that experience a similar environment. When the receivers become transmitters themselves and eventually get acknowledgement for a successful transmission, a reward calculation routine based on this idea is invoked. Transmitting stations scan the $\Sigma CW[]$ vector, calculate the frequencies (popularity) of CW values appearing there, by $\frac{counter_{CW}}{length(\Sigma CW[])}$ and store the results in a vector $Frequencies_{CW}[7]$ which has a size dictated by the different possible CW values. This vector then gets sorted in descending order, while the algorithm keeps track of what index (CW value) corresponds to which frequency. The agent gets rewarded depending on the order the CW size it used for that transmission has in that vector.

The agent rewards itself more for using CW values that are placed first in order on that vector (are often used to successfully transmit a packet), and less for CW values that are near the end of the vector (are rarely used), by employing equally distanced rewards. This way, the reward function just considers the order of CW levels by their popularity, but not the popularity itself ($\frac{counter_{CW}}{length(\Sigma CW[])}$) so that it is more fair and the Q-Learning agent does not get biased early on and fixed on a potentially wrong CW trajectory. Specifically, when the transmitting station succeeds (and gets an ACK) using the most commonly successful (popular) CW size within its first hop neighbours with same transmission properties (no exploratory packets, similar data rate), its embedded Q-Learning agent is given the maximum possible reward. Every other CW placing below that in order of popularity will get its acquired reward reduced by 1/7th at a time (since we consider 7 CW levels). i.e. in the case of the least popular CW (with the least successful transmissions in the near network), the reward multiplier will be 1/7. The mechanism's operation is summarised in Algorithm 2.

The CCE reward function is expected to improve fairness and reduce the convergence time of the Q-Learning algorithm, thus give a bigger performance benefit earlier. It is also quite efficient regarding networking overhead since it costs just 3 bits per packet to represent the 7 CW levels which can be easily absorbed by the IEEE 802.11p standard. It could also be adapted for prioritisation among different classes of data since many proposed techniques use different CW sizes for the same purpose.

2) Combination of two sub-goals: Similar logic regarding reward assignments can be applied to introduce delay awareness to the protocol, as seen in [18]. As mentioned, the CW parameter is defined as the number of timeslots the station has to weight prior to transmitting, so the smaller this parameter is, the better in terms of total latency. The smaller CW values can be given higher reward. The larger the CW size, the lower the reward given.

Additionally we can further optimise performance, by combining the two objectives (fairness and low latency). This can be achieved by specifying even more detailed reward function, featuring 49 discrete reward levels (equally distanced from each other) if the proposed fairness-aware, CCE reward function is used in conjunction with a delay-aware reward function. This would also focus the agent on a trajectory even faster than using just 2 or 7 reward levels as shown before.

$$R_{func}(CW) = R_{CCE}(CW) \times R_{delay}(CW) \quad (5)$$

We found the approach in (5) to be more efficient when it comes to minimising latency than a "softer" reward approach of combining rewards like $R_{func}(CW) = \frac{R_{CCE}(CW)+R_{delay}(CW)}{2}$, via which the agent can receive relatively high rewards without necessarily achieving a high reward from both the delay-aware and CCE functions. So i.e., the reward would be $r_t \leftarrow r_{CCE} \times r_{delay} = 1/7 \times 4/7 = 4/49$ for using the CW value which is the least common found in receipt packets, but is averagely favourable for delay intolerant applications. Effectively, using the product of the result of the two functions as a reward, makes the one act as a filter to the other. This way, the agent is less punished when it simultaneously achieves both sub-goals (low latency, fairness) in a single transmission. If the designer of a system needs to add bias towards one optimisation factor compared to the other, a weighted product function can be used as follows,

$$R_{func}(CW) = R_{CCE}^{k_{CCE}}(CW) \times R_{delay}^{k_{delay}}(CW) \quad (6)$$

where the weights, $k_{CCE} + k_{delay} = 2$ and $0 < k_{CCE}, k_{delay} < 2$. The neutral case in (5) will be given for $k_{CCE} = k_{delay} = 1$. A schematic of the protocol's operation utilising both enhancements (fairness and delay awareness) is seen below in Fig. 4.

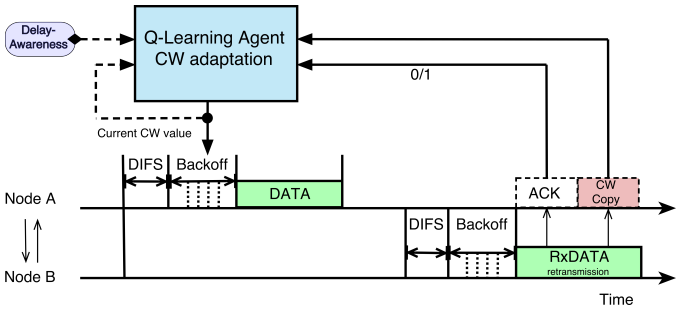

Fig. 4: Q-Learning MAC with fairness and latency optimisations

V. Performance Evaluation

A MAC protocol should achieve three main objectives when the wireless medium is shared among multiple vehicle stations; bandwidth efficiency, low latency, and fairness. Consequently, we evaluate our designs against that criteria.

A. Experiment Setup

We perform our tests in a simulated vehicular environment with moving IEEE 802.11p stations implemented with OMNeT++ 5 and the Veins framework. The SUMO mobility co-simulator takes care of the vehicle movement aspect. All vehicles are placed on a 3-lane highway and travel with a maximum velocity of 15 m/s so that the maximum distance travelled is 4.5 km in 300 s. The Krauss mobility model is used with default parameters as seen in Table I, and the maximum distance among them reaches up to no more than 1 km as the simulation progresses. A snapshot of the formation of vehicles at 100 s of simulation time can be seen in Fig. 5. The source code of our simulation featuring examined protocols under all scenarios is made available[1] for further reference.

The Veins IEEE 802.11p implementation does not support unicast transmissions at the time of writing. We can emulate unicast with probabilistic retransmissions as non-blocking ACKs originating from the application layer, as suggested in [19]. These retransmissions can be used for forwarding purposes as well. Veins focuses on broadcast, OCB IEEE 802.11p, which does not feature ACKs. Consequently, the IEEE 802.11p Veins implementation does not feature the BEB part of the DCF, since it relies on explicit ACK packets to adjust the $backoff$ parameter depending on whether a transmission was successful or not. For the purpose of comparison, we implemented a Pseudo-BEB mechanism based on feedback

[1]https://github.com/apressas/omnet_qmac_vanet

Parameter	Value
Simulation time	300 s
ε-decay (training) period	180 s
Channel Frequency	5.89 GHz
Channel Bandwidth	10 MHz
Transmission rate R	9 Mbit/s
Transmission power	Single-hop: 30 dBm Multi-hop: 17 dBm
Packet size L_p	256 bytes
Backoff slot time	13 μs
Packet Generation Frequency f_{gen}	10 Hz
Packet Generation Offset	0.005 s
Discount rate γ	[0.7-0.9]
Mobility Model	Krauss model with default parameters ($\sigma = 0.5$, $\tau = 1$)
Maximum Vehicle Velocity	15 m/s
Vehicle Acceleration Ability	2.6 m/s^2
Vehicle Deceleration Ability	4.5 m/s^2

TABLE I: Simulation Parameters

from non-blocking, application-layer ACKs on top of the IEEE 802.11p Veins implementation.

By using the proposed IEEE 802.11p with probabilistic retransmissions, we can have feedback regarding the outcome of transmissions in a broadcast environment. This means that the proposed algorithms can be applied for purely unicast transmissions but they also comply with the IEEE 802.11p specification which primarily operates in OCB mode to allow one-to-many information exchanges.

Most proposed V2V applications need a packet transmission frequency of at least 10 Hz [28], while some need even up to 50 Hz [29]. In our simulations, every station generates 10 original packets/s, and also retransmits original packets received from others with a variable probability P_{fwd}. Some asynchronisation is introduced to transmissions by adding a randomised offset time that can reach a maximum of 0.005 s. The retransmitted packets carry acknowledgements that are needed for reliability purposes as well as feedback for MAC mechanisms. In practice, an acknowledgement can be carried by any broadcasted packet, since most of the payload would still be utilised to enable other applications. In our implementation, they are just replicas of messages, so that we can collect fair measurements when approaching channel saturation. They are also used for forwarding purposes in multi-hop deployments.

All packets have a common header which is similar to Cooperative Awareness Messages (CAMs) or Decentralized Environmental Notification Messages (DENMs), but is modified to include Node ID, application type, whether a packet is original or a retransmission, the employed CW and whether that CW was used due to exploration or exploitation. We do not deal with QoS-enabled MAC architectures, where links could have different priorities, but our design can also be adapted to manage contention among different services in a way similar to the IEEE 802.11 EDCA mechanism, which also uses different CW values to separate them regarding urgency.

1) Simulation Parameters: The scenarios envisaged in this work consider $N_{vehicles} = 50$ or 100 stations; each station

Fig. 5: The 3-lane highway scenario used in network simulations. Green/red colours of vehicles identify successful/unsuccessful transmission of their latest generated packet respectively.

generates data packets with constant rate $f_{gen} = 10\,\text{Hz}$ by employing a bit rate, R, which would depend on the experienced channel quality. The receivers can calculate the forwarding (ACK) probability P_{fwd} in real time from (7),

$$P_{fwd} = P_{ACK} = \frac{N_{ACK}}{N_{vehicles}} \tag{7}$$

by detecting the number of relevant nearby active transmitters via the incoming packets containing the node IDs and the number of hops, so as to consider only immediate neighbours and disregard packets received from multi-hop paths (retransmissions). We can get the maximum theoretical network-wide throughput from the following equation,

$$T_h = N_{vehicles} \times f_{gen} \times (1 + N_{ACK}) \times L_p \times 8\,bit \tag{8}$$

which gives us 3.072 Mbit/s for 50 transmitting stations and 6.144 Mbit/s for 100 transmitting stations, sending $L_p = 256\,\text{byte}$ packets with $N_{ACK} = 2$, which is chosen so that an ACK will be received with higher confidence since the packet delivery probability in studied systems is less than 1. We set $R = 9\,\text{Mbit/s}$ so that the channel does not bottleneck even the denser scenario, while it can conveniently accommodate more than a 100 vehicles within the one-hop range [30].

Regarding Q-Learning training and evaluation, the discount factor γ is in the range of 0.7 to 0.9. The learning rate α and ε-decay, training function (4) lasts for 180 s or $N_{train} = 1800\,(5400)$ original (total) packets, with $\lambda = 3$. As mentioned, we expect this method of training to behave better in real deployments since it forces the agent to explore all action-state pairs early on, and then focus on the most promising trajectory. The evaluation stage starts when the ε-greedy function reaches constant $\varepsilon = 0.05$ and lasts for 120 s. We present 5-minute snapshots of the agent's behaviour under various configurations and metrics, that combine both the training and evaluation stages.

2) Benchmarked Protocols:

- **IEEE 802.11p:** It is the baseline protocol operating in OCB (broadcast) mode with fixed $CW = CW_{min} = 3$, as defined in the standard for the fastest AC. It has no CW adaptation capability.
- **Pseudo-BEB:** The addition of retransmissions originating from the receivers' application layer allowed us to emulate the BEB algorithm for the IEEE 802.11p MAC and compare our novel Q-Learning protocols against it in a fully broadcast, OCB system, as well as emulate unicast transmissions.
- **Q_MAC:** Our original protocol first presented in [27] with a binary reward function. Its operation is shown in Fig. 3.
- **Q_MAC+CCE:** Our novel protocol introduced in this paper based on Q-Learning in conjunction with the CCE reward algorithm where $R_{func} = R_{CCE}$
- **Q_MAC+Delay:** It is the Q-Learning agent using a delay-aware reward function where $R_{func} = R_{delay}$.
- **Q_MAC+Delay+CCE:** A novel protocol which targets satisfying both sub-goals, utilising (5).

Applying a moving average filter to the CW recordings over time reveals the mean system-wide CW over time. From these CW dynamics, we can make interesting observations about the significance of this parameter in dense IEEE 802.11p networks, as well as evaluate the collective behaviour of the Q-Learning agents over time using various reward functions. It can be seen in Fig. 6 that all the proposed solutions try to minimize the medium congestion level by enforcing different CW values on communications.

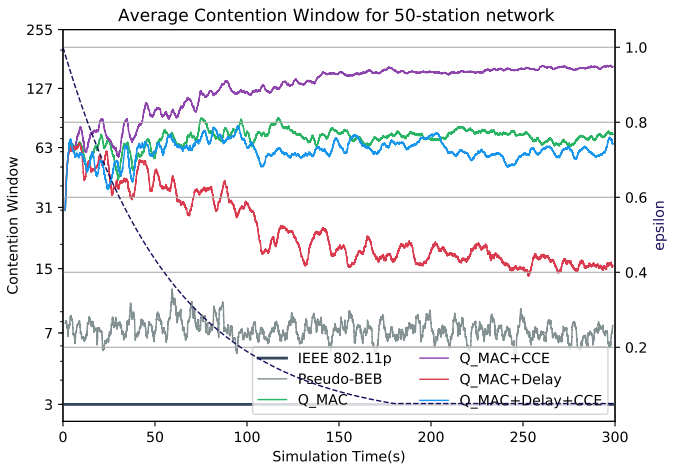

Fig. 6: System-wide CW dynamics for different collision avoidance mechanisms

The original Q-Learning MAC protocol strives for maximum transmission reliability, and the one with the CCE reward function strives for both reliability and fairness regarding contention. The delay-aware function tries to use a CW as small as possible while achieving acceptable reliability. When combining both reward functions, as in (5), the mean system-wide CW is quite higher since the agent strives for reliability and fairness, but still lower than the other two Q-Learning based solutions.

3) Performance Metrics: Our goal is to study the intrinsic fairness properties of the IEEE 802.11p DCF and the proposed MAC mechanisms, so first we concentrate on homogeneous

singe-hop scenarios in which all stations experience similar transmission conditions, meaning that no station is disadvantaged by its signal quality, traffic pattern, or spatial position, or other asymmetries. Then we evaluate the performance of the proposed approaches on a multi-hop scenario which is subject to the phenomena mentioned above.

We use receiver-centric metrics to evaluate the performance of the suggested approaches since they better represent the level of awareness every vehicle has of its surrounding vehicles. Raw throughput in terms of intact packets received over time is measured at all receivers and then a moving average filter is applied so that we can collect a system-wide reading over time. That way the real-time effect of the learning algorithm onto network performance can be evaluated.

On the other hand, end-to-end latency of received transmissions is measured at a single station placed in the middle of the network. Each generated packet contains the time it was created at the application layer, and is subtracted by the time of reception by the receiving node's application layer. We consider only packets while $\varepsilon = \alpha = 0.05$ to properly assess the results of Q-Learning algorithms post-convergence.

The fairness objective can be characterized in two different manners: long-term and short-term. Long-term fairness is measured over long time periods, corresponding to the transmission of many packets by a station, i.e., 1000 or more. A MAC protocol is considered to be long-term fair if the probability of successful channel access observed over a long period of time (many packets transmitted) converges to $1/N$ for N competing hosts. But a MAC protocol should also provide equal opportunity for access to the medium over short time periods as well, i.e., lasting a few seconds or tens of packets transmitted per station. A MAC protocol can be long term fair but short-term unfair, meaning that one host may continuously capture the channel over short time intervals. Vehicles transmit safety-related, irreplaceable packets with a short time of relevance. All cars should be given equal transmission opportunities, not only in the long term but in the short term as well (i.e., 2-4 s - the duration of 20 to 40 original or 60 to 120 total transmissions/station).

$$J(x_1, x_2, ..., x_n) = \frac{(\Sigma_{i=1}^{n} x_i)^2}{n \cdot \Sigma_{i=1}^{n} x_i^2} \tag{9}$$

We calculate Jain's fairness index [31] shown in (9), which is a popular metric for measuring the unfairness of an allocation vector, and has also been used to evaluate VANETs' fairness before [32]. We adopt it for analysing the fairness of achieved throughput among wireless vehicular stations. The index value equals unity corresponds to the fairest allocation in which all stations achieve the same throughput. We set the fairness criterion to be $J = 95\%$, according to other works dealing with fairness in IEEE 802.11-based systems, such as [33] [34]. The number of received packets from all transmitters are measured at a single vehicle and J is calculated over a sampling window of 1 to 10 s (short-term to long-term) with a step of 0.5 s. This result is averaged over equally spaced starting points with $\varepsilon = \alpha = 0.05$, to obtain smoother and more accurate graphs.

In the following, we show our findings regarding throughput, fairness and latency in four different V2V scenarios.

B. Medium Traffic Environment

We first evaluate the proposed and existing MAC protocols when deployed in a VANET of 50 vehicles transmitting 256-byte packets. In Fig. 7, it can be seen how each protocol utilises the channel, since efficient bandwidth usage is their primary objective. The protocols' performance is evaluated against the maximum achievable throughput which is found via performing an exhaustive search among the available CW values applied globally (to all stations in a network simultaneously). The CCE reward function (Q_MAC+CCE) clearly performs better regarding achieved throughput, since the agents collectively estimating congestion do a better job than every one acting completely independently. Also the use of similar CW sizes is enforced, which leads to less collisions. The other Q-Learning based solutions also perform quite better than the baseline OCB IEEE 802.11p with $CW = CW_{min}$. Our BEB implementation on the other hand is not yielding a great increase in throughput compared to the original protocol. The poor performance of BEB is due to the increase of collisions under increased network traffic load, since the mechanism is collision-triggered and resets a station's CW to CW_{min} after every successful transmission. On the other hand, the proposed solutions update the CW around a value that resolves as many collisions as possible and keep it there.

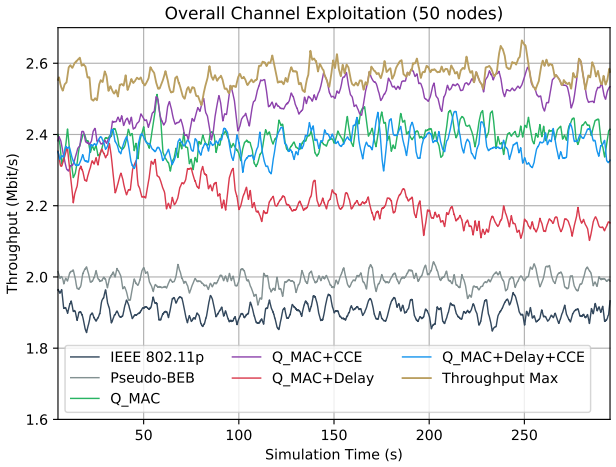

Fig. 7: Mean network-wide throughput for 50 stations

Furthermore, the achieved transmission latency is examined. The Normalised Cumulative Distribution Function (CDF) is produced, that shows raw latency recorded over percentage of successfully received packets for each protocol, shown in Fig. 8. An interesting observation from Fig. 8 is that each solution shows different performance limits on delay and packet deliver ratio. With a more relaxed delay deadline, non-delay sensitive solutions show better packet delivery ratio, e.g., achieving maximum throughput can translates to almost 79% of packet delivery ratio but with a latency of up to 40 ms. Q_MAC+CCE is very close at 77%, and outperforms the

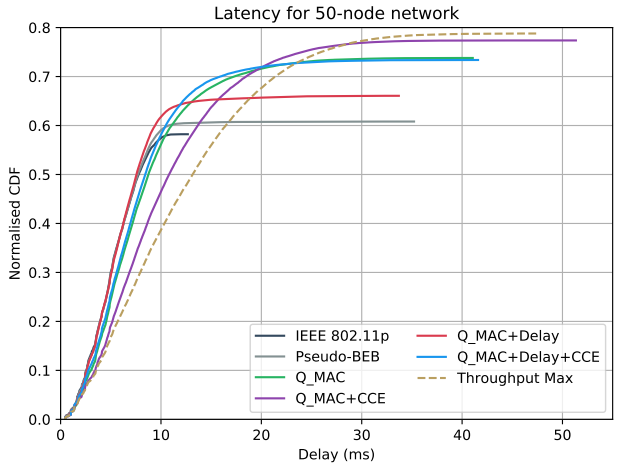

Fig. 8: End-to-end transmission latency in 50-station network

maximum throughput solution for latency requirements below 30 ms.

Additionally, given latency requirements of 12 ms to 20 ms, our Q_MAC+Delay+CCE performs better than the rest of the protocols achieving the highest transmission reliability, i.e., a packet delivery ratio of 72% shown on the Y-axis. Q_MAC+Delay is the best solution if latencies lower than 12 ms are needed. So we conclude that with appropriate tuning (balancing the trade-off between delay awareness and CCE with (6)), the Q-Learning MAC protocol can better satisfy even the most stringent delay requirements for the medium-traffic environment. This figure can be used as a guideline to select a suitable access strategy given an application requirement.

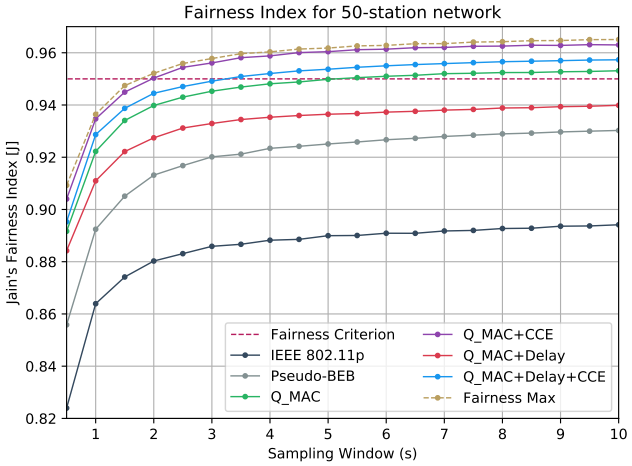

Fig. 9: Fairness for 50 stations

When it comes to fairness measurements, shown in Fig. 9, the CCE enhanced Q-Learning agents perform better than the simpler protocols they are based on (Q_MAC and Q_Delay), as expected. Specifically, the Q_MAC+CCE protocol meets our strict fairness criterion even within 2 s or 60 packet transmission attempts per station, (compared to 4.5-5 s for Q_MAC) which favours critical exchanges. If the CCE reward function is combined with delay-awareness (Q_MAC+Delay+CCE), the same level of packet-based fairness is achieved within 3 s. The delay-focused reward function without CCE performs quite worse in that regard, since it does not meet the fairness criterion (J>0.95) even in a 10 s sample - or 300 transmitted packets per station. We conclude that for this sparser scenario, using the proposed CCE reward function makes a significant difference regarding fair bandwidth allocation among vehicles.

C. High Traffic Environment

We then test 100 contenting stations transmitting 256-byte packets. Aggregate throughput measurements over time for the system are shown in Fig. 10. When compared to each other, the protocols perform as in the previous scenario regarding achieved throughput. Although the performance gap between the proposed CCE reward function and maximum is slightly wider, i.e., 6.33% during 300 s of simulation time, given more time, the protocol can achieve maximum throughput. In terms of the practical requirements on short-term performance and applicability in VANETs, the algorithm can yield the presented gain over time or be pre-trained and activated in dense environments where there is big quantity of information to be exchanged among vehicles tuned in the same channel.

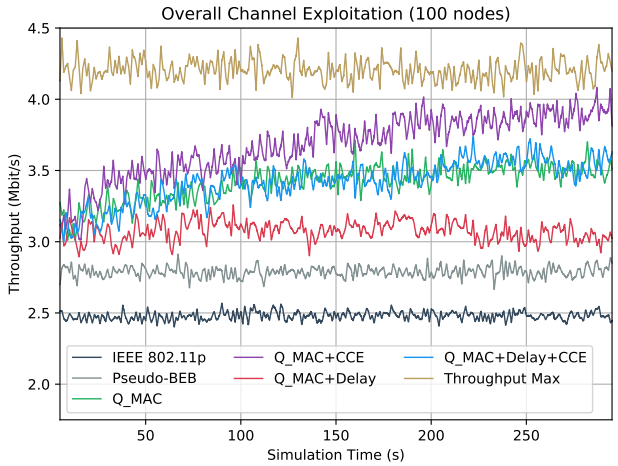

Fig. 10: Mean network-wide throughput for 100 stations

But when comes to transmission latency, shown in Fig. 11, the learning MAC with joint CCE and delay-awareness outperforms all MAC solutions in terms of packet delivery for latency requirements among 22 to 33 ms. Q_MAC, which cannot be further controlled performs quite closely. The Q_MAC+Delay protocol, which defines what is possible when focusing on low latency exchanges, outperforms the rest for 13.5 to 22 ms. Given a delay requirement of 100 ms which is typical for V2V applications, Q_MAC+CCE is more preferable in practice since it achieves the highest delivery ratio.

When it comes to transmission fairness, shown in Fig. 12 the results align with the ones collected from the first scenario. Both CCE-enhanced Q-Learning protocols are throughput-fair within shorter time than their non-CCE counterparts. The BEB

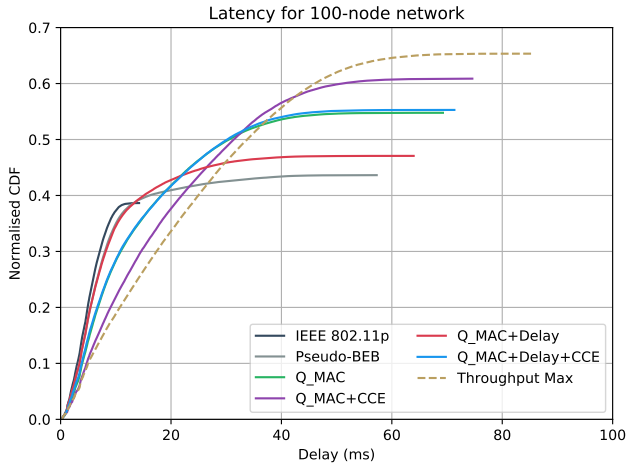

Fig. 11: End-to-end transmission latency in 100-station network

and Q_MAC+Delay are not fair in the short term or long term when evaluated against our criterion. The baseline DSRC MAC also cannot handle 100 cars regarding neither long-term nor short-term fairness.

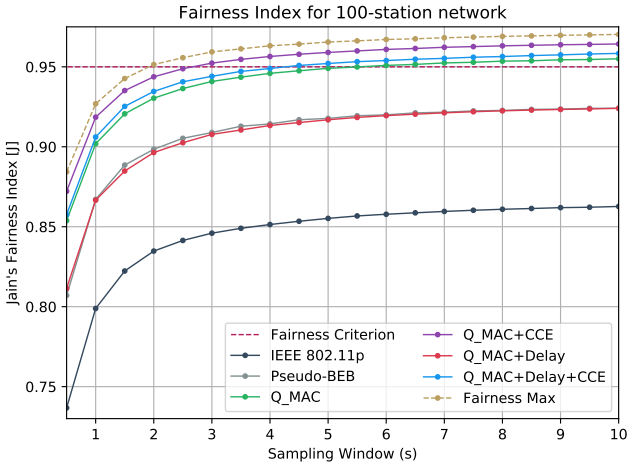

Fig. 12: Fairness for 100 stations

D. Two simultaneous services

The same mechanism for improving fairness on a network level can be employed by the protocol to better accommodate multiple simultaneous applications, by the same (EDCA-like priorities) or different stations. We enforce application separation regarding CW by making the CCE algorithm check the application type field which is contained in the packets, meaning that only CW values from packets of the same application get copied and affect the Q-Learning reward function. Additionally, only stations running the same application retransmit each other's packets so that we can collect fair measurements.

We simulate stations of two types, running different application layers. To make a fair comparison regarding raw network-wide throughput, we set 80% of the vehicles to transmit 256-byte packets and 20% of the vehicles to transmit 1024-byte packets. Consequently in the scenario of 50 vehicles presented below, 40 cars run the first application and 10 cars run the second one. Assuming no packet losses, the throughput of the two applications should be equal to each other ($\frac{T_{hB}}{T_{hA}} = 1$). Only stations running the same application collectively estimate the optimum application-wide CW, instead of all stations trying to find the optimum system-wide CW. The recorded application-wide throughput can be seen in Fig. 13 and Fig. 14 for applications A and B respectively.

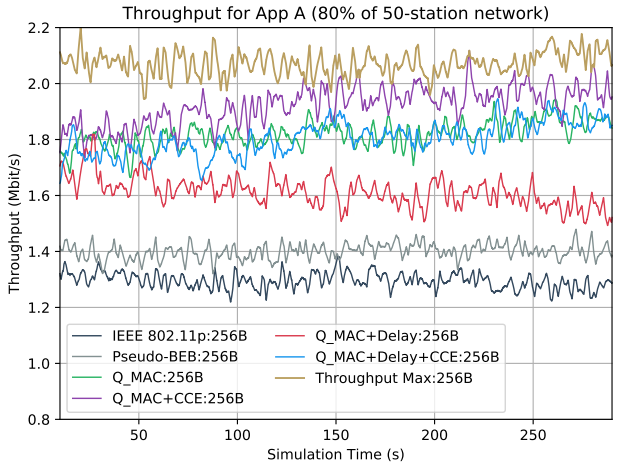

Fig. 13: Total throughput achieved by stations sending 256-byte packets

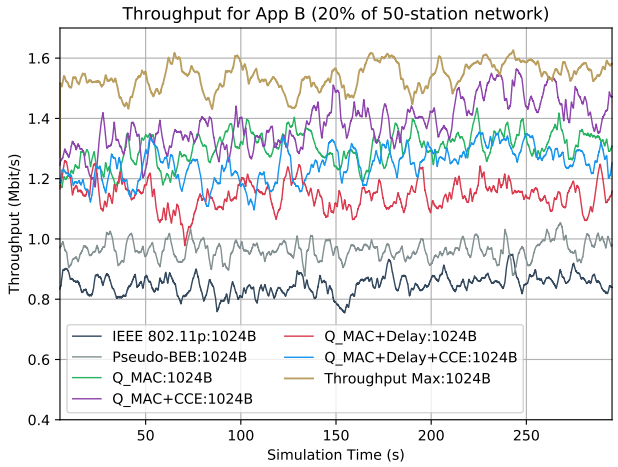

Fig. 14: Total throughput achieved by stations sending 1024-byte packets

It can be observed that there is significant increase in throughput (approaching the maximum) when our novel learning technique is applied to the DSRC MAC. Although the throughput of the two applications would be equal should there

be no contention, in practice bigger packet transmissions are more prone to collisions, and if losses occur the throughput is affected more because of the larger packet size. This is reflected in the collected results, as expected. But if we evaluate application-wide fairness expressed as a ratio of throughput of application B (1024-byte packets) over application A (256-byte packets), the proposed learning technique shows significant improvement over the DSRC stack. The Q_MAC+CCE protocol achieves a ratio of up to $\frac{T_{hB}}{T_{hA}} \approx 0.74$ for throughput of application B over application A, compared to 0.658 for the baseline IEEE 802.11p solution, while yielding the highest overall throughput as well, within 6.5% of the maximum.

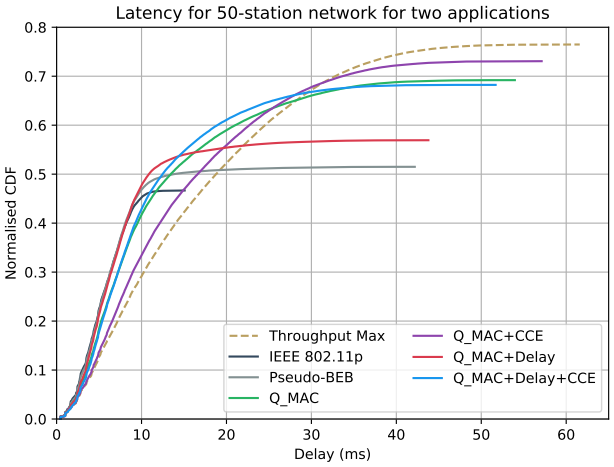

Fig. 15: End-to-end transmission latency in 50-station network for 2 applications

Regarding the end-to-end latency of successful transmissions, depicted in Fig. 15, we observe that using R_{func} with both sub-goals combined (Q_MAC+Delay+CCE), favours low latency exchanges, with the protocol achieving the higher delivery ratio for latencies below 28 ms all the way down to 14 ms end-to-end. Again, given a delay requirement of 100 ms which is typical for most V2V applications, the protocol with the highest raw throughput Q_MAC+CCE performs better.

Network-wide fairness for all stations, no matter the application they are running, can be seen in Fig. 16. IEEE 802.11p with or without the BEB exhibits a more severe fairness problem under these multi-rate conditions, which can be tackled using the learning-based methods. We can again confirm that resetting to CW_{min} harms delivery and fairness performance under sustained high traffic. Again, better performance can be achieved when the proposed CCE method is utilised in conjunction with the Q-Learning MAC, with or without delay awareness. The fairness aware learning protocol Q_MAC+CCE achieves $J = 95\%$ within a window of 2-2.5 s or 60-75 transmitted packets per station, while Q_MAC achieves the same of fairness within about 6.5-7 s or 195-210 packets. The Q_MAC+Delay+CCE protocol can reach the set criterion within 3.5 s for this simulation scenario, while the basic latency-optimised protocol without the CCE function (Q_MAC+Delay) cannot reach the fairness criterion at all.

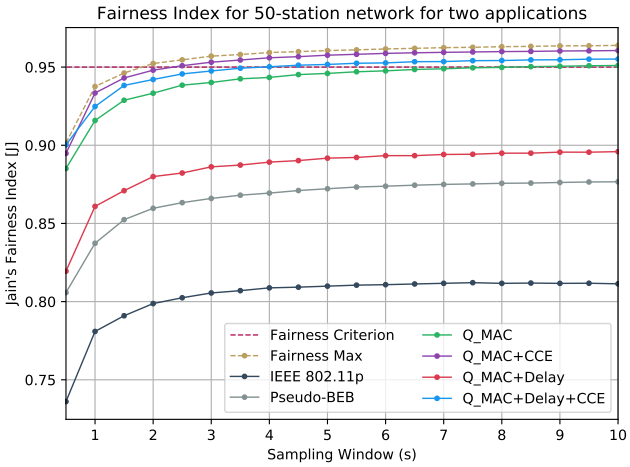

Fig. 16: Fairness for 50 stations running 2 applications

E. Multi-Hop Environment

The performance of the Q-Learning MAC has also been studied under a multi-hop network environment of 100 stations, which are placed at most 2 hops away from each other. To achieve that, transmission power is lowered, as seen in Table I. Every vehicle periodically calculates its packet forwarding probability P_{fwd} depending on the number of its one-hop neighbours via eq. (7), by setting $N_{fwd} = N_{ACK} = 6$ to ensure coverage for the given RoI, even in the increased presence of collisions because of hidden terminals. Each vehicle forwards a copy of a received packet at most once to limit redundancy. Again the Q_MAC+CCE protocol yields the highest raw throughput among the protocols, as seen in Fig. 17. It can also be observed that it learns how to increase performance faster than the rest of the protocols.

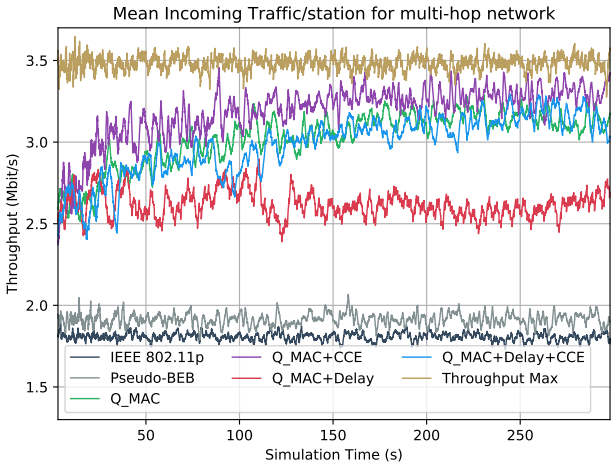

Fig. 17: Experienced incoming traffic in multi-hop network

When it comes to latency performance in this scenario, depicted in Fig. 18, only unique copies of packets are considered, whether they come from single-hop or two-hop paths, since this reveals more about the performance of the system.

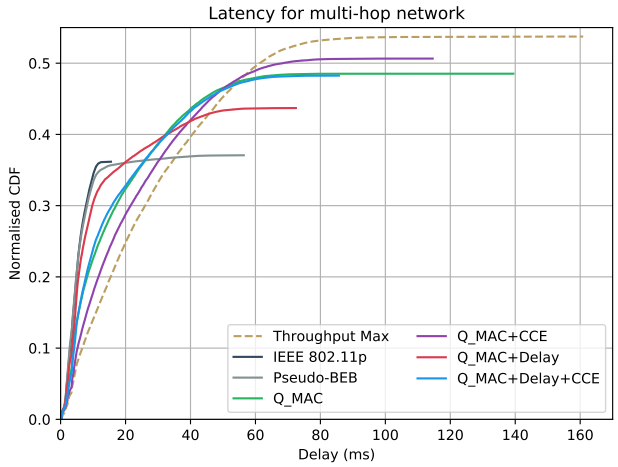

Fig. 18: End-to-end transmission latency in multi-hop network

Combining CCE and delay awareness in the reward function (Q_MAC+CCE+Delay) with equal bias yields better performance for requirements among 34.5 ms to 47.5 ms, very close to that of Q_MAC which cannot be further controlled. As always, biasing the protocol towards delay with $k_{delay} > k_{CCE}$ in (6) can yield even higher delivery rates for latency-sensitive transmissions. Focusing entirely on delay (Q_MAC+Delay) will make the Q-Learning algorithm outperform all the rest for latencies down to 19 ms in a multi-hop setting. For a more common multi-hop transmission requirement of 100 ms the Q_MAC+CCE again yields the highest performance.

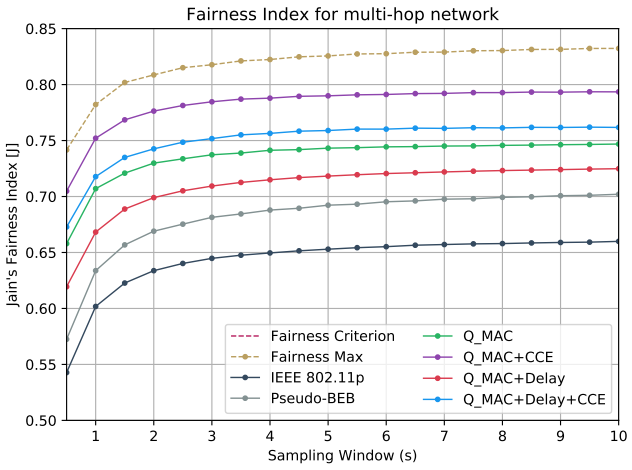

Fig. 19: Fairness among 100 flows in multi-hop network

We also evaluate fairness among flows of unique packets considering their origin (the vehicle that generated the packet) and disregarding whether they arrive via single-hop (the vehicle that generated the information) or multi-hop (forwarding vehicle) paths. This way we can assess the performance of the multi-hop network regarding its capability to fairly carry information among all vehicles in the RoI, whether they are immediate (single-hop) neighbours of the receiver or not. Results are depicted in Fig. 19. Achieved multi-hop fairness is naturally lower, but CCE-enhanced protocols continue to vastly improve on the simpler Q-Learning protocols they are based on, with the best performing just 5% below the maximum fairness measured for the system. Q_MAC+CCE can reach $J = 79.4\%$, compared to the simpler Q_MAC with the binary reward function which goes up to $J = 74.6\%$. Similarly, Q_MAC+Delay+CCE goes up to $J = 76.2\%$, while Q_MAC+Delay can reach a maximum of $J = 72.45\%$ within 10 s or 100 original packets transmitted per vehicle.

VI. Conclusion

We found that compared to the base IEEE 802.11p CSMA and BEB protocols, the Q-Learning MAC protocol can largely mitigate the collision problem in congested VANETs by discovering the appropriate CW value to be used for transmissions. Additionally, CCE-enhanced Q-Learning MAC protocols consistently outperform the protocols they are based on, in terms of fairness and raw throughput. When combining both CCE and delay-awareness mechanisms, designers can bias the Q-Learning agent towards either high delivery for delay-sensitive traffic or strive for maximum data rates for large exchanges. So there is a clear trade-off when biasing the learning agent; it can strive towards maximum raw throughput and fairness or reliable low-latency transmissions, or a combination of the two, depending on the given application.

The reward function presented in this work can be used to trade raw throughput and fairness for lowering transmission latency or the opposite. For the presented evaluation, the edge and equal-bias cases of the reward function in eq. (6) are tested. More combinations of biasing the reward function could be tested in future work to examine how the protocol responds to tuning, towards enabling applications that would require reliable low latency exchanges, while achieving some acceptable level of fairness. Combining Q-Learning with a sliding CW technique to enable EDCA-like priorities for differentiating network traffic depending on requirements could also be an interesting future study.

References

[1] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in dsrc," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '04, (New York, NY, USA), pp. 19–28, ACM, 2004.

[2] Y. Wang, A. Ahmed, B. Krishnamachari, and K. Psounis, "Ieee 802.11p performance evaluation and protocol enhancement," in *2008 IEEE International Conference on Vehicular Electronics and Safety*, pp. 317–322, Sept 2008.

[3] S. Eichler, "Performance evaluation of the ieee 802.11p wave communication standard," in *2007 IEEE 66th Vehicular Technology Conference*, pp. 2199–2203, Sept 2007.

[4] X. Wang and G. B. Giannakis, "Csma/cca: A modified csma/ca protocol mitigating the fairness problem for ieee 802.11 dcf," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, p. 039604, Mar 2006.

[5] G. Naus, R. Vugts, J. Ploeg, R. v. d. Molengraft, and M. Steinbuch, "Cooperative adaptive cruise control, design and experiments," in *Proceedings of the 2010 American Control Conference*, pp. 6145–6150, June 2010.

[6] Z. Xu, X. Li, X. Zhao, M. H Zhang, and Z. Wang, "Dsrc versus 4g-lte for connected vehicle applications: A study on field experiments of vehicular communication performance," *Journal of advanced transportation*, vol. 435, 08 2017.

[7] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "Macaw: A media access protocol for wireless lan's," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 212–225, Oct. 1994.

[8] Z. Li, A. Das, A. K. Gupta, and S. Nandi, "Performance analysis of ieee 802.11 dcf: Throughput, delay, and fairness," *Unpublished. Available on 16th Mar*, 2011.

[9] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet of Things Journal*, vol. 1, pp. 289–299, Aug 2014.

[10] C. Campolo, A. Vinel, A. Molinaro, and Y. Koucheryavy, "Modeling broadcasting in ieee 802.11p/wave vehicular networks," *IEEE Communications Letters*, vol. 15, pp. 199–201, February 2011.

[11] C. Campolo, A. Molinaro, A. Vinel, and Y. Zhang, "Modeling prioritized broadcasting in multichannel vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 687–701, Feb 2012.

[12] B. Kloiber, J. Härri, T. Strang, and S. Sand, "Bigger is better — combining contention window adaptation with geo-based backoff generation in dsrc networks," in *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 227–233, Nov 2014.

[13] M. I. Hassan, H. L. Vu, and T. Sakurai, "Performance analysis of the ieee 802.11 mac protocol for dsrc safety applications," *IEEE transactions on vehicular technology*, vol. 60, no. 8, pp. 3882–3896, 2011.

[14] R. Reinders, M. van Eenennaam, G. Karagiannis, and G. Heijenk, "Contention window analysis for beaconing in vanets," in *2011 7th International Wireless Communications and Mobile Computing Conference*, pp. 1481–1487, July 2011.

[15] Q. Wu, S. Nie, P. Fan, Z. Li, and C. Zhang, "A swarming approach to optimize the one-hop delay in smart driving inter-platoon communications," *CoRR*, vol. abs/1807.07301, 2018.

[16] S. Amuru, Y. Xiao, M. van der Schaar, and R. M. Buehrer, "To send or not to send - learning mac contention," in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2015.

[17] Z. Liu and I. Elhanany, "Rl-mac: A qos-aware reinforcement learning based mac protocol for wireless sensor networks," in *2006 IEEE International Conference on Networking, Sensing and Control*, pp. 768–773, April 2006.

[18] C. Wu, S. Ohzahata, Y. Ji, and T. Kato, "A mac protocol for delay-sensitive vanet applications with self-learning contention scheme," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pp. 438–443, Jan 2014.

[19] F. Klingler, F. Dressler, and C. Sommer, "Ieee 802.11p unicast considered harmful," in *2015 IEEE Vehicular Networking Conference (VNC)*, pp. 76–83, Dec 2015.

[20] M. Impett, M. S. Corson, and V. Park, "A receiver-oriented approach to reliable broadcast in ad hoc networks," in *2000 IEEE Wireless Communications and Networking Conference. Conference Record (Cat. No.00TH8540)*, vol. 1, pp. 117–122 vol.1, Sept 2000.

[21] T. Li, T. Tang, and C. Chang, "A new backoff algorithm for ieee 802.11 distributed coordination function," in *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 3, pp. 455–459, Aug 2009.

[22] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[23] C. J. C. H. Watkins, *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, May 1989.

[24] M. J. Mataric, "Reward functions for accelerated learning," in *Machine Learning Proceedings 1994*, pp. 181–189, Elsevier, 1994.

[25] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[26] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[27] A. Pressas, Z. Sheng, F. Ali, D. Tian, and M. Nekovee, "Contention-based learning mac protocol for broadcast vehicle-to-vehicle communication," in *2017 IEEE Vehicular Networking Conference (VNC)*, pp. 263–270, Nov 2017.

[28] B. Bilgin and V. Gungor, "Performance comparison of ieee 802.11 p and ieee 802.11 b for vehicle-to-vehicle communications in highway, rural, and urban areas," *International Journal of Vehicular Technology*, vol. 2013, 2013.

[29] J.-M. Lee, M.-S. Woo, and S.-G. Min, "Performance analysis of wave control channels for public safety services in vanets," *International Journal of Computer and Communication Engineering*, vol. 2, no. 5, p. 563, 2013.

[30] US National Public Safety Telecommunications Council, "5.9 ghz dsrc operational concept introduction."

[31] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *CoRR*, vol. cs.NI/9809099, 1998.

[32] W. Alasmary and O. Basir, "Achieving efficiency and fairness in 802.11-based vehicle-to-infrastructure communications," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pp. 1–6, May 2011.

[33] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau, "Fairness and its impact on delay in 802.11 networks," in *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, vol. 5, pp. 2967–2973 Vol.5, Nov 2004.

[34] G. Berger-Sabbatel, A. Duda, M. Heusse, and F. Rousseau, "Short-term fairness of 802.11 networks with several hosts," in *Mobile and Wireless Communication Networks* (E. M. Belding-Royer, K. Al Agha, and G. Pujolle, eds.), (Boston, MA), pp. 263–274, Springer US, 2005.

Andreas Pressas received the M.Eng degree in Electrical & Electronic Engineering from the department of Engineering & Design of the University of Sussex. He is currently a Ph.D researcher with the same department. His current research interests include IoT and vehicular ad hoc networks protocols and applications, including medium access control protocol design and applications of computational intelligence for performance enhancement of such networks.

Zhengguo Sheng received the B.Sc. degree from University of Electronic Science and Technology of China, in 2006 and M.S. and Ph.D. degrees from Imperial College London, UK, in 2007 and 2011, respectively. He is currently a Senior Lecturer with University of Sussex, UK. Previously, he was with UBC as a Research Associate and with Orange Labs as a Senior Researcher. He has more than 100 publications. His research interests cover IoT, vehicular communications, cloud/edge computing.

Falah Ali received the B.Sc. degree in electrical and electronics engineering and the M.Sc. degree in electronic systems from Cardiff University, in 1984 and 1986, respectively, and the Ph.D. degree in communications engineering from the University of Warwick, in 1992. He is currently a Reader in digital communications and the Director of Communications Research Group. His research interests include multiuser and multiantenna communication techniques, vehicular communications and WSNs.

Daxin Tian is a professor in the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include mobile computing, intelligent transportation systems, vehicular ad hoc networks, and swarm intelligence.